(NASA-CR-184071) THE SOFTWARE PRODUCT ASSURANCE METRICS STUDY: JPL'S SOFTWARE SYSTEMS QUALITY AND PRODUCTIVITY (Jet Propulsion Lab.) 77 p CSCL 09B

N89-25618

Unclas

G3/61 0219935

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. JPL Pub. 89-6	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle The Software Product Assurance Metrics Study: JPL's Software Systems Quality and Productivity		5. Report Date February 15, 1989 6. Performing Organization Code
7. Author(s) Marilyn W. Bush		8. Performing Organization Report No.
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109		10. Work Unit No. 11. Contract or Grant No. NAS7-918 13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This document reports the findings of the Jet Propulsion Laboratory (JPL)/Software Product Assurance (SPA) Metrics Study, conducted as part of a larger JPL effort to improve software quality and productivity. Until recently, no comprehensive data had been assembled on how JPL manages and develops software-intensive systems. The first objective of this study was to collect data on software development from as many projects and for as many years as possible. Results from five projects are discussed. These results reflect 15 years of JPL software development, representing over 100 data points (systems and subsystems), over a third of a billion dollars, over four million lines of code and 28,000 person months. Analysis of this data provides a benchmark for gauging the effectiveness of past, present and future software development work. In addition, the study is meant to encourage projects to record existing metrics data and to gather future data. The SPA long term goal is to integrate the collection of historical data and ongoing project data with future project estimations. If we don't know where we have been and where we are now, it is impossible to assess the productivity and quality of future software development projects and systems.

17. Key Words (Selected by Author(s)) Quality Assurance and Reliability Computer Programming and Software Operations Research Statistics and Probability Systems Analysis	18. Distribution Statement		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 69	22. Price

JPL 0184 R 9/83

JPL Publication 89-6

The Software Product Assurance Metrics Study: JPL's Software Systems Quality and Productivity

Marilyn W. Bush

February 15, 1989

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

This document reports the findings of the Jet Propulsion Laboratory (JPL)/Software Product Assurance (SPA) Metrics Study, conducted as part of a larger JPL effort to improve software quality and productivity. Until recently, no comprehensive data had been assembled on how JPL manages and develops software-intensive systems. The first objective of this study was to collect data on software development from as many projects and for as many years as possible. Results from five projects are discussed. These results reflect 15 years of JPL software development, representing over 100 data points (systems and subsystems), over a third of a billion dollars, over four million lines of code and 28,000 person months. Analysis of this data provides a benchmark for gauging the effectiveness of past, present and future software development work. In addition, the study is meant to encourage projects to record existing metrics data and to gather future data. The SPA long term goal is to integrate the collection of historical data and ongoing project data with future project estimations. If we don't know where we have been and where we are now, it is impossible to assess the productivity and quality of future software development projects and systems.

ACKNOWLEDGMENTS

Thanks to the Software Product Assurance Section data collection team: Yaakov Ben-Ami, Billa Dahaby, Michael Huang, Allen Nikora, Mitch Scaff, Steve Tyler, Gina Walton, Cheryl White, and Richard Yeh. Faiza Lansing, the original team leader, helped put much of the data and analysis together. Yaakov Ben-Ami, John Kelly and Allen Nikora helped refine the analysis. Lynn Gresham and Valerie Mathews helped shepherd the report through last-minute details and revisions.

Mark Kennedy, Fikry Lansing, Harry Ohls, Bill Paine, and Tommy Thompson were liaisons in the gathering and interpretation of data. Tara Douglas and Mary Parrish typed the report, and Neal Palmer polished the text.

Al Pietrasanta helped put the report on track and made sure it stayed there.

Finally, the report could not have been completed without several hundred project personnel who went out of their way to supply needed data. It is, after all, their work that is recorded here.

CONTENTS

1. INTRODUCTION 1

1.1 PURPOSE AND SCOPE 1
1.2 BACKGROUND 1
1.3 STUDY METHOD 2
1.4 OUTCOMES 2

2. DATA COLLECTION METHODOLOGY 4

2.1 PROCESS 4
2.2 SOURCES OF DATA 5
2.3 DATA COLLECTION PROBLEMS 8

3. RESULTS 9

3.1 PRODUCTIVITY 10
3.1.1 PRODUCTIVITY DATA 10
3.1.2 PRODUCTIVITY RATIOS 12
3.1.3 PRODUCTIVITY CORRELATIONS 14
3.1.4 COMPARABLE PRODUCTIVITY DATA . . . 19

3.2 QUALITY 22
3.2.1 QUALITY DATA 22
3.2.2 QUALITY RATIOS 24
3.2.3 COMPARABLE QUALITY DATA 26

4. CONCLUSIONS 28

5. FUTURE PLANS 29

BIBLIOGRAPHY 30

APPENDIX A DEFINITIONS A-1
APPENDIX B VOYAGER B-1
APPENDIX C GALILEO C-1
APPENDIX D SFOC D-1
APPENDIX E NON-NASA E-1
APPENDIX F DSN MARK IV F-1

FIGURES AND TABLES

Figure

1 FLIGHT SOFTWARE COST 15

2 FLIGHT SOFTWARE WORK EFFORT 16

3 GROUND SOFTWARE COST 17

4 GROUND SOFTWARE WORK EFFORT 18

B.1 THE VOYAGER PROJECT SYSTEM CONFIGURATION . . B-4

C.1 THE GALILEO PROJECT SYSTEM CONFIGURATION . . C-4

D.1 THE SFOC PROJECT SYSTEM CONFIGURATION D-4

F.1 THE DSN MARK-IV DSCC SYSTEM COMPONENTS . . . F-4
F.2 THE DSN MARK-IV NOCC SYSTEM COMPONENTS . . . F-5
F.3 THE DSN MARK-IV GCF SYSTEM COMPONENTS F-6
F.4 THE DSN MARK-IV SPT SYSTEM COMPONENTS F-7

Table

1 AVAILABLE COSTS (DOLLARS AND WORKYEARS) 7

2 COMPREHENSIVE PRODUCTIVITY DATA 11

3 PRODUCTIVITY RATIOS 13

4 COMPARABLE PRODUCTIVITY DATA 21

5 COMPREHENSIVE QUALITY DATA 23

6 QUALITY RATIOS 25

7 COMPARABLE QUALITY DATA 27

B.1 VOYAGER PERIOD 1 - PRELAUNCH B-8
B.2 VOYAGER PERIOD 2 - LAUNCH TO SATURN B-9
B.3 VOYAGER PERIOD 3 - SATURN TO URANUS B-10

C.1 GALILEO PRELAUNCH PHASE C-6

D.1 SFOC D-7

F.1 DSN MARK-IV F-9

SECTION 1

INTRODUCTION

1.1 Purpose and Scope

This document reports the findings of the JPL Software Product Assurance (SPA) Metrics Study. The SPA Metrics Study was conducted as part of a larger effort to generate quality and productivity metrics (measurements) for software development at JPL and ultimately to improve software quality and improve the productivity and predictability of software development.

The first objective of the SPA Metrics Study was to collect data on software development from as many projects and for as many years as possible. Analysis of this data provides a benchmark for gauging the effectiveness of past, present and future software development work. (If you do not know where you have been and where you are, it is difficult to know where you are going.) As an added benefit, the study was meant to encourage projects to record existing metrics data and systematically to gather future data.

What follows will discuss results from five projects: Voyager, Galileo, the Space Flight Operations Center (SFOC), a non-NASA project, and the Deep Space Network (DSN) Mark IV.

1.2 Background

JPL is now and will be developing software-intensive systems on a large scale. But up to now, JPL has had no way to determine exactly what it does well and what needs improvement. Until recently no comprehensive data existed on how JPL manages and develops software-intensive systems.

Over the years, bits and pieces of data have been collected by individuals on various projects. However, the type of data collected varied according to the interests of the person collecting it. If an effort was not made soon, data from past projects would be lost.

Over the last five years, the JPL directors have expressed concern about the way the Laboratory develops and manages software-intensive systems. In 1985, JPL's Deputy Director commissioned a Software-Intensive System Study. In February 1986, the study team came up with a set of conclusions and recommendations. One was to create the Systems Software and Operations Resource Center (SSORCE), formed later in 1986, whose primary function is to formulate standards for developing and managing JPL software systems. In 1987, the Software

Product Assurance (SPA) Section was chartered to improve the software development and management process.

In July 1987, as part of its charter, SPA began collecting and analyzing JPL project data with the objective of creating Lab-wide measures of quality and productivity for software development.[1] This report represents the first attempt to assemble this data.

1.3 Study Method

Collecting the data was like putting pieces of a puzzle together. To assemble the data SPA staff began talking with project people, especially those who had been interested in the kind of data SPA needed and those who could refer SPA to others who had the data. In all, over 150 JPL people were interviewed, ranging from project managers, software managers, system engineers, and subsystem engineers to cognizant programmers and cognizant development engineers. Actual cost and staffing information was readily available from the project offices participating in the study. Software development costs, however, were more difficult for project offices to identify. In these cases, SPA staff were referred to the appropriate system and subsystem engineers and managers.

Old memorandums and reports were examined as well as old financial planning histories, work breakdown structures, and configuration management plans. Information sources varied according to each project's propensity to keep information.

Data was assembled based on four basic parameters: lines of code, dollars, workmonths, and defects. (See Appendix A for the definition of each parameter.) Using these four basic parameters, quality and productivity baselines were constructed. **Quality** is defined as **the number of defects per thousand lines of source code. Productivity** is defined in two ways: **dollars per source line of code**, and **source lines of code per workmonth.**

1.4 Outcomes

At the beginning of the study it appeared that quantitative data on JPL's software quality and productivity might be virtually impossible to retrieve because it had not been kept in any standardized form. The study showed, however, how it could (with difficulty) be gathered.

---

[1]In March 1988, the Systems Software and Operations Resource Center, through the Software Product Assurance Resource Center, contributed funds to help carry on this work. (Their funding amounts to about 25 percent of the total funding for this study.)

This effort has made it possible to refine the collection of future metrics data. It has also raised the project staffs' awareness of the value of keeping accurate records in a standard format. Most importantly, it has laid the foundation for developing cost-estimating models and project tracking guidelines for the JPL environment.

SECTION 2

DATA COLLECTION METHODOLOGY

2.1 Process

Data collection proceeded in the following steps:

(a) Identify the project. Outline project systems and subsystems.

(b) Develop a set of questions relating the four basic parameters to the project's systems and subsystems.

(c) Meet with the project management staff and determine the information available in the project office. Identify other sources of information.

(d) Interview the system engineers and managers and any other individuals recommended by the project office. During these interviews SPA staff:

  (1) Corroborated the information obtained from the project office.

  (2) Corrected and clarified information obtained from the project office. In cases where information about a system was provided both at the project and system levels, it was assumed that data from the system managers and engineers was more accurate.

  (3) Added the system/subsystem engineers' information to the information from the project office.

  (4) Identified other sources of data.

(e) Contact the project configuration management organization and institutional failure reporting centers. These organizations helped obtain line-of-code and failure-count data. (Information gathered from configuration management organizations varied from project to project. Some project configuration management centers kept a record of the number of lines of code delivered for each subsystem while others did not.)

(f) Interview the subsystem task managers, cognizant engineers and

any other individuals identified by the system managers and engineers:

(1) Corroborate information obtained from the project office and system managers and engineers.

(2) Correct and clarify information gathered from these sources. Where information about a subsystem was provided at the project, system, and subsystem levels, it was assumed that data from the subsystem task managers and engineers was more accurate.

(3) Add new information gathered from the subsystem task managers and cognizant engineers to the SPA database.

(4) Identify further sources of information.

(g) Analyze the overall quality of incoming information. Preliminary estimates were produced for the cost of a new line of code delivered to operations and of the number of observed defects per thousand lines of code at the subsystem level. The SPA staff then compared these results to numbers from comparable JPL project systems and subsystems.

(h) Check the validity of these computations by reviewing them with system and subsystem engineers and project management staff. Since many individuals had not seen information in this form before, such reviews were often useful. Project staff would remember previously undiscussed aspects of the work and would frequently be able to help SPA revise its original estimates. (On one occasion, for example, Voyager's ground data system engineer remembered costs that weren't included in our numbers and identified costs that should not have been included. This helped refine SPA's productivity figures for two subsystems.)

2.2 Sources of Data

Although many people were interviewed, SPA staff tried in all cases to go back to the original sources of data. For **lines of code**, this could mean old memorandums, release description documents, configuration management library reports, monthly management reviews, etc. For information about **dollars** and **workyears**, the study used financial planning history: B805, C805, D805, Resources Status Reports (RSRs), work breakdown structures, etc. Failure reports, software problem reports, discrepancy and anomaly reports, software change requests, etc. were examined for **defect** data. (Software change requests, however, were not included in the defect count because they did not report failures.)

The study collected data from as many original sources as were

available and corroborated the information with the appropriate project personnel. Sometimes more than one source contributed to a single piece of data. These sources, documents and people, are listed in the Appendices for each project.

Table 1 lists the cost information (dollars and workyears) made available to the study by each project. From this, it can be seen that for all projects except SFOC the system engineering costs were included. Management costs were included only for the non-NASA project. Software development costs were included for all projects. Contract management costs were included for Voyager Periods 1, 2, 3, Galileo, the non-NASA project, and partially for DSN Mark IV. No spacecraft testing or operations were included for any project. Product assurance costs were only included for the non-NASA project and in part for DSN Mark IV. System integration costs were included for all projects except SFOC and DSN Mark IV.

Table 1. Available Costs (Dollars and Workyears)

Project	System Eng	Mgmt	SW Develop	Contract Mgmt	Space-craft Test	Opera-tions	Product Assurance	System Integra-tion
Voyager-Period 1	Yes	No	Yes	Yes	No	No	No	Yes
Voyager-Period 2	Yes	No	Yes	Yes	No	No	No	Yes
Voyager-Period 3	Yes	No	Yes	Yes	No	No	No	Yes
Galileo	Yes	No	Yes	Yes	No	No	No	Yes
SFOC*	No	No	Yes	No	N/A	No	No	No
Non-NASA*	Yes	Yes	Yes	Yes	N/A	No	Yes	Yes
DSN-Mark IV**	Yes	Partial	Yes	Partial	N/A	No	Partial	No

Projects SFOC, non-NASA and DSN-Mark IV have ground systems only. Spacecraft testing is not applicable.
* Projects SFOC and non-NASA are in the development stages.
**Cost and WY are only partial info. No WY data available before FY82.

2.3 Data Collection Problems

As with all studies, there were problems. Many of the original project personnel were no longer available or had forgotten details of past projects. In addition, data had been tracked differently on different projects.

One major benefit of gathering quantitative data is that it established what kinds of data were ordinarily collected or not collected by various projects. JPL's budget and account system provides records of dollars and workyears, but it is not always easy to relate these to specific software subsystem efforts. Keeping track of lines of code data presented several discrepancies: new code versus old code, source code versus object code, estimated lines of code versus actual lines of code.

In collecting **defect** data, the numbers represent a lower bound of actual defects. Many defects probably went unreported and those that were reported were classified differently across projects. Although defects may occur throughout the development lifecycle, this study has focused on the most troublesome kind: post-development defects. **Unless otherwise indicated, the word "defect" in this report always refers to a post-development defect.** Failure reports are missing for a number of subsystems. In some cases software failure reports were combined with hardware failure reports, making the reports unusable. Also, there are still no laboratory-wide defect definitions.

For **source lines of code**, again no standard laboratory definition exists nor any overall reporting mechanism. Therefore, with some exceptions, this information was pieced together based on what the cognizant engineers remembered.

For **dollars** and **workyears** it was difficult to distinguish between development costs and related costs (e.g., configuration management costs). It was also hard to relate subsystem costs to the phases of the lifecycle. And the use of the SRM/RSR system as a reporting mechanism made deciphering the data difficult. (It was hard to distinguish software development costs from hardware development costs, management costs from development costs, etc.) Under this system, contractor labor can be listed as a procurement cost, along with equipment of all types. This further complicated data gathering.

SECTION 3

RESULTS

No one involved with this metrics project anticipated how comprehensive it would grow in less than a year, especially considering the relatively modest level of personnel and funding available. The present database encompasses:

- 102 data points, covering the systems and subsystems of 5 projects:

Flight Systems	Data Points
Voyager Period 1 (Development to Launch)	4
Voyager Period 2 (Launch to Saturn)	4
Voyager Period 3 (Saturn to Uranus)	3
Galileo	3
Subtotal	14
Ground Systems	
Voyager Period 1	7
Voyager Period 2	7
Voyager Period 3	7
Galileo	7
SFOC	11
Non-NASA Project	18
DSN Mark IV	31
Subtotal	88
TOTAL	102

- 4,851,274 source lines of code
- 28,311 workmonths
- $366,862,000 dollars

To put these numbers in perspective, the study surveyed the equivalent of 236 people working for ten years, each producing 20,556 lines of code, and spending a total of a third of a billion dollars.

3.1 Productivity

3.1.1 Productivity Data

Table 2 shows the productivity data, summarized by project, and separated into flight systems and ground systems. The basic parameters of productivity--lines of code, dollars, and workmonths--are itemized for each project.

As can be seen from Table 2, there is a good spread of projects, from small to large. The four flight systems range from 1,200 source lines of code (SLOC) to 27,790 SLOC, and the seven ground systems range from 45,300 SLOC to 1,288,144 SLOC. Similar ranges occur in the dollar and workmonth parameters, both for flight and ground systems.

Table 2. Comprehensive Productivity Data*

Project	SLOC	$K**	Workmonths
Flight Systems			
Voyager Period 1	13,950	18,682	1,202
Voyager Period 2	1,200	1,874	158
Voyager Period 3	1,200	1,302	198
Galileo	27,790	28,861	2,786
Subtotal	44,140	50,719	4,344
Ground Systems			
Voyager Period 1	742,195	46,938	3,545
Voyager Period 2	178,300	5,970	851
Voyager Period 3	45,300	3,247	371
Galileo	1,278,911	50,095	4,633
SFOC ***	342,224	11,322	1,195
Non-NASA ***	829,180	108,788	4,752
DSN MARK IV	1,288,144	89,783	8,620
Subtotal	4,704,254	316,143	23,967
TOTAL	**4,748,394**	**366,862**	**28,311**

*Adjusted totals. See tables in Appendices B through F for details.

**FY '87 Dollars, with the exception of SFOC (1986-1988 real dollars) and Non-NASA (1984-1988 real dollars)

***On-Going

3.1.2 Productivity Ratios

Table 3 presents productivity ratios that can be derived from the data in Table 2 and shows average productivity for all analyzed projects over the last 15 years.

For flight systems it has cost an average of $1,149 to produce one line of new source code. Ten lines of source code were produced per average workmonth. For ground systems, on the average, a line of new source code cost $67, with 186 lines of source code produced per workmonth.

If these productivity ratios seem outrageously low, one should remember that, for some of the projects, far more than "programming" is included in these figures. All the workmonths of all the software professionals from the start of a project through its delivery to system test are being counted. Actual programming constitutes only a small percentage of a total system development effort.

Now for the first time the relative productivity of flight and ground systems can be compared. A flight systems line of source code costs seventeen times more than a ground systems line of source code. Put another way, ground systems can produce nineteen times as many lines of source code per workmonth as flight systems.

Engineers experienced in the development of these systems have recognized significant productivity differences between flight systems and ground systems (although the 17-to-1 range may surprise some). Flight systems must be designed to fit in very limited onboard spacecraft computer memory. The extreme reliability and performance requirements of onboard code necessitate design, reviews, simulation, testing, etc., sufficient to support a degree of reliability not normally required for ground software.

Table 3. Productivity Ratios

Project	Dollars*/ SLOC	SLOC/ Work Month
Flight Systems		
Voyager Period 1	1339	12
voyager Period 2	1562	8
Voyager Period 3	1085	6
Galileo	1039	9
Cumulative Average:***	$1149	10
Ground Systems		
Voyager Period 1	63	209
Voyager Period 2	33	210
Voyager Period 3	72	122
Galileo	39	276
SFOC**	33	286
Non-NASA**	131	123
DSN Mark IV	70	149
Cumulative Average:***	$67	186

*FY '87 Dollars, with the exception of SFOC (1986-1988 real dollars) and non-NASA (1984-1988 real dollars)
**On-Going

*** Adjusted total dollars for all projects divided by adjusted total SLOC for all projects.
Adjusted total SLOC for all projects divided by adjusted total workmonths for all projects.
See tables in Appendices B through F for details of adjusted totals.

3.1.3 Productivity Correlations

To determine the strength of the relationship between these productivity ratios, simple regression analyses were performed on each set of subsystem data. The results are displayed in Figures 1 through 4.

For Figure 1: Flight software dollars ($K) vs. SLOC:

$K = 2135.13 + 0.71 SLOC, R = 0.70

For Figure 2: Flight software workmonths vs. SLOC:

WM = 126.73 + 0.07 SLOC, R = 0.79

For Figure 3: Ground software dollars ($K) vs. SLOC:

$K = 862.97 + 0.04 SLOC, R = 0.74

For Figure 4: Ground software workmonths vs. SLOC:

WM = 85.20 + 0.003 SLOC, R = 0.71

FIGURE 1

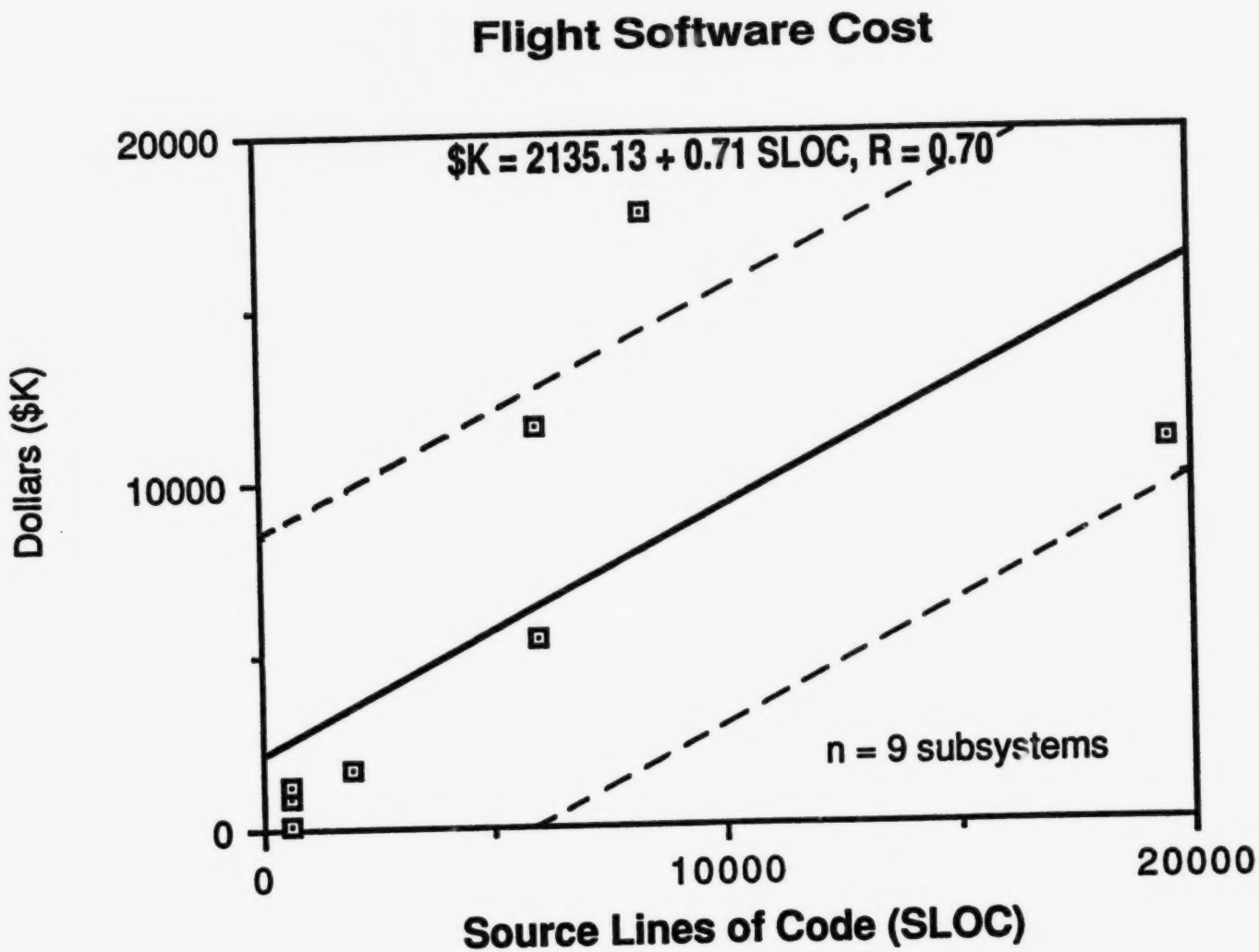

Flight Software Cost
$K = 2135.13 + 0.71 SLOC, R = 0.70
n = 9 subsystems
Dollars ($K)
Source Lines of Code (SLOC)
20000
10000
0
0
10000
20000

FIGURE 2

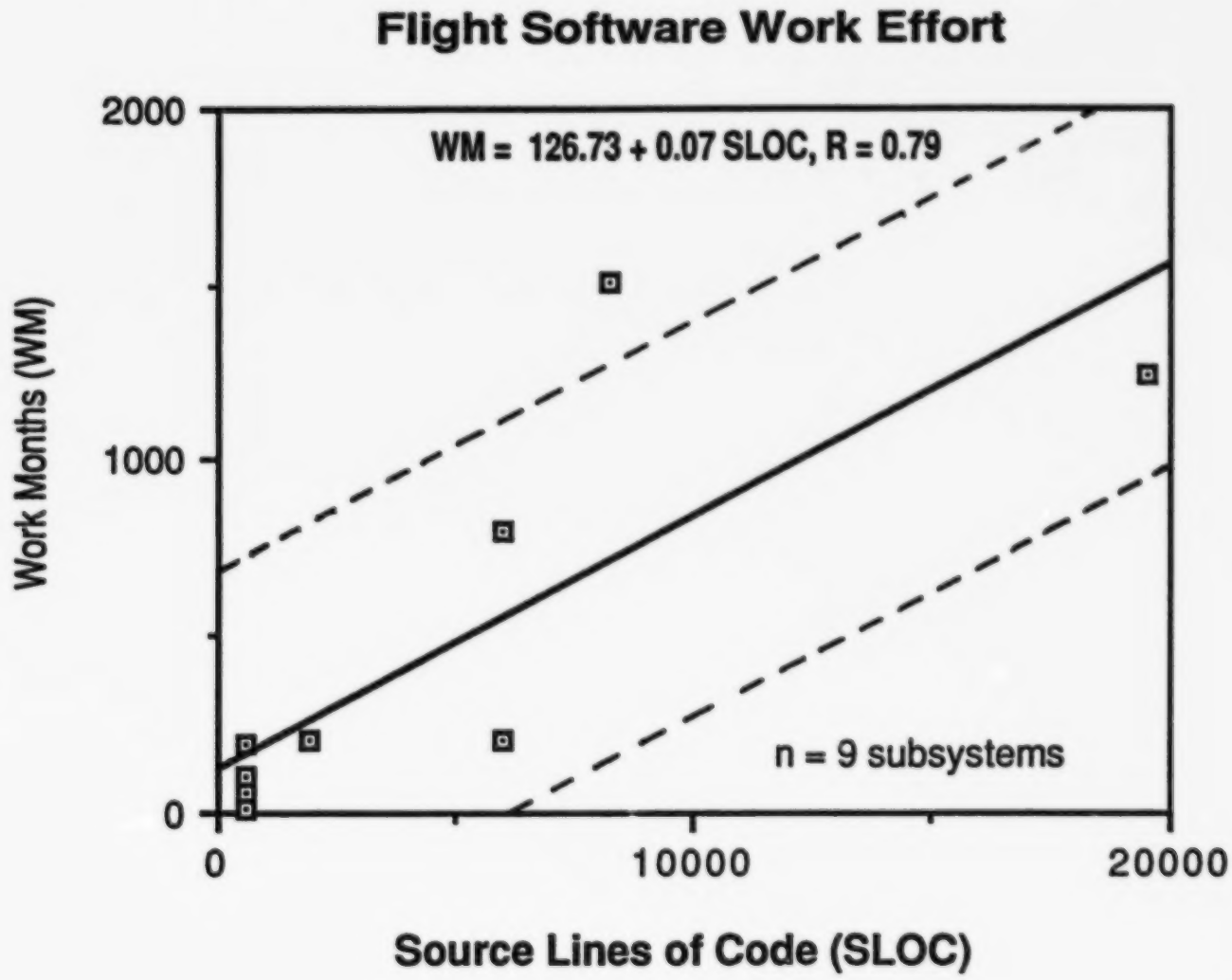

Flight Software Work Effort
WM = 126.73 + 0.07 SLOC, R = 0.79
n = 9 subsystems
Work Months (WM)
2000
1000
0
0
10000
20000
Source Lines of Code (SLOC)

FIGURE 3

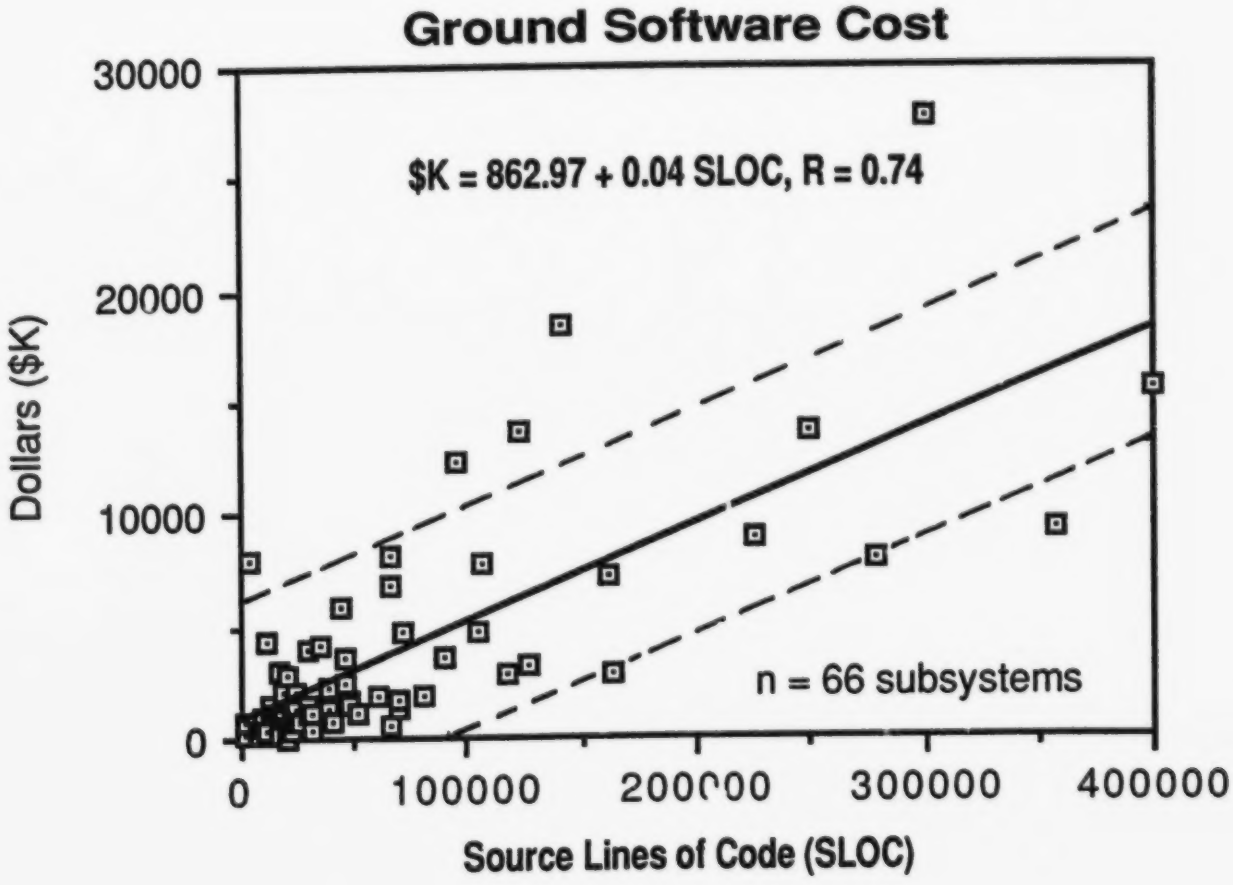
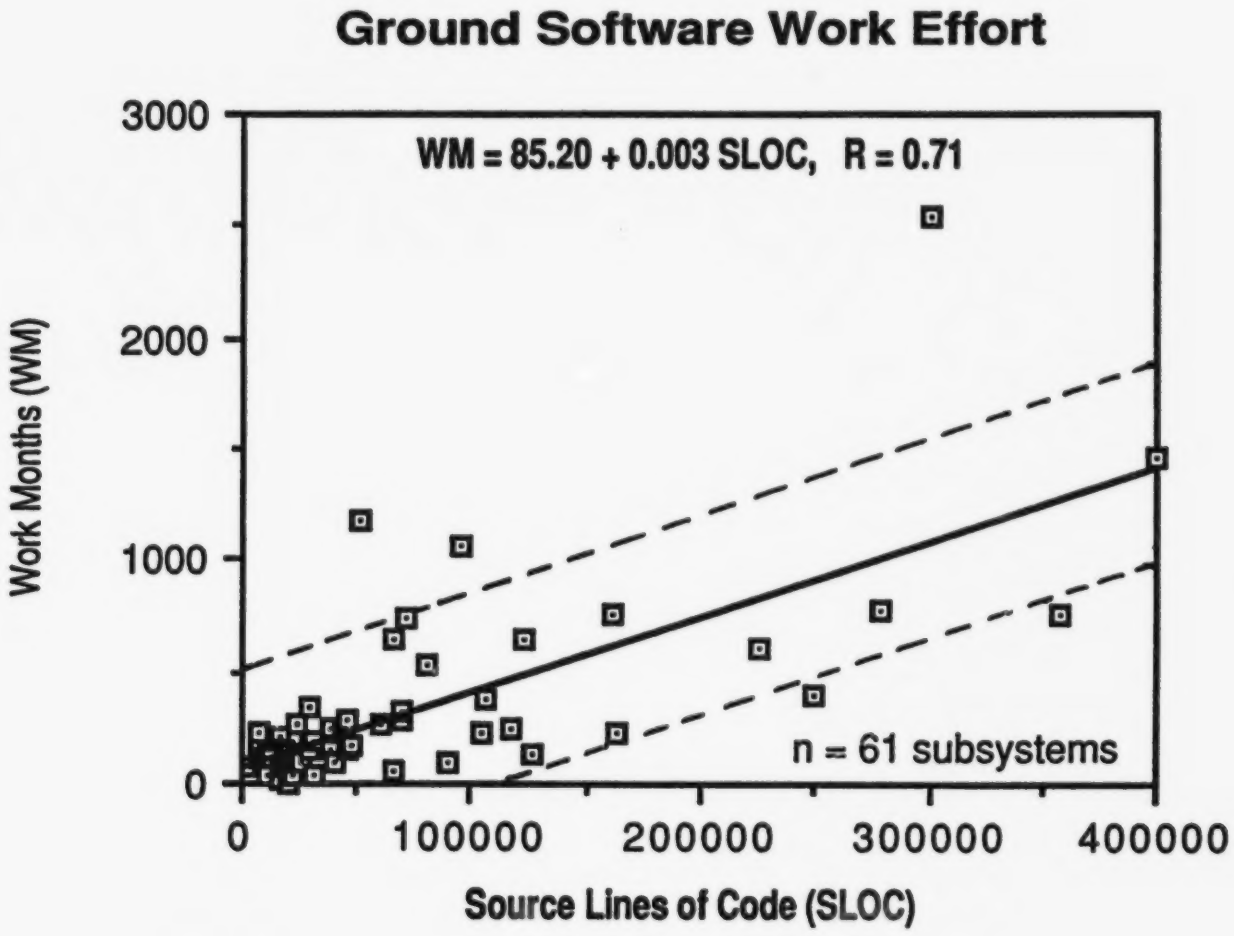

Ground Software Cost
$K = 862.97 + 0.04 SLOC, R = 0.74
n = 66 subsystems
Dollars ($K)
Source Lines of Code (SLOC)
30000
20000
10000
0
0
100000
200000
300000
400000

FIGURE 4


Ground Software Work Effort
WM = 85.20 + 0.003 SLOC, R = 0.71
n = 61 subsystems
Work Months (WM)
Source Lines of Code (SLOC)
3000
2000
1000
0
0
100000
200000
300000
400000

On all four figures a simple linear regression is used and error bars (two dashed lines) represent one standard deviation.

Given that the data was derived from a variety of sources several years old, SPA staff were surprised to find relatively high correlation coefficients of 0.70 through 0.79. These correlations indicated a stronger relationship than was previously expected from JPL archival data.

One conclusion implied by the correlations is that the cost of producing different kinds of flight software and ground system software is relatively consistent, even when the systems themselves vary in size. This gives one encouragement that productivity ratios can be successfully used as one technique to estimate costs of new projects. For example, if one line of source code for flight systems costs $1,149, a 10,000 line of source code flight system should cost around $11,490,000 (in FY'87 value dollars), and take around 1000 workmonths.

3.1.4 Comparable Productivity Data

Given the productivity results for a large database of JPL projects, how do these ratios compare with productivity data from similar non-JPL projects? Table 4 presents three other productivity studies: one involves ten non-JPL NASA projects; a second, the IBM Houston Space Shuttle project; and the third, the Magellan ground software project at Martin-Marietta.

In all three studies, the ground software figures are close to the JPL ratios. The SLOC-per-Workmonth numbers for the NASA projects (130-260) and for the Magellan project (191) compare with JPL's 186. The cost of a ground SLOC for the Space Shuttle project ($95-125) compares with JPL's $67 (Table 3).

For flight software, the SLOC-per-Workmonth productivity figures for the NASA projects (44-88) and for the Space Shuttle project (40-80) are higher than JPL's figure (10). It will take further evaluation to determine whether this is due to different ways of counting, errors in the data, or the possibility that JPL flight systems are intrinsically more difficult to produce than the other measured systems. A final possibility is that development methodologies of flight software projects elsewhere might be more productive than JPL's.[2]

---

[2]On the other hand, a recent Air Force study showed productivity averages of 10 to 20 SLOC per workmonth for flight systems, about the same as JPL's (personal conversation with Bob Guarino of Tecolote Research, Inc.).

In summary, the productivity data stands up to two methods of corroboration: there appears to be internal data consistency, as shown by the high correlations; there also appears to be external consistency, as shown by comparison to non-JPL projects.

Table 4. Comparable Productivity Data

Average JPL productivity	
- Flight Software	10 SLOC/Workmonth
- Ground Software	186 SLOC/Workmonth
Productivity of 10 Non JPL NASA Projects*	
- Flight Software	44-88 SLOC/Workmonth
- Ground Software	130-260 SLOC/Workmonth
Space Shuttle, IBM FSD Houston**	
- Flight Software	40-80 SLOC/Workmonth $500/SLOC (50% maintenance)
- Ground Software	$95-125/SLOC (50% maintenance)
Magellan (Partials), Martin Marietta***	
- Ground Software	191 SLOC/Workmonth

* Hamid Habib-Agahi, James Quirk, Shantanu Malhotra, "Software Productivity and Costs in NASA Projects," JPL 900-990, January, 1988 (JPL internal document)

** Unpublished report

***Paul Scheffer, Allen Bucher, "Software Productivity on a Portion of the Magellan Spacecraft Ground Data System," June 1987 (Martin Marietta Astronautics Group of Martin Marietta Denver Aerospace, under contract #956700 with JPL)

3.2 Quality

3.2.1 Quality Data

Table 5 presents quality data for four flight projects (the three periods of Voyager and Galileo) and for six ground projects. The quality data incorporates SLOC data and adds defect data. The defect data derives from a variety of failure reports (see Section 2.2) and only shows post-development defect numbers.

SFOC and the non-NASA project are ongoing projects and therefore are still in the process of finding defects. In the case of the non-NASA project, post-development defects are just beginning to be found and are not included in this report. For SFOC, there is data on development and post-development defects. The post-development defect numbers for these projects represent a definite lower bound (true for all projects in this report) and can be expected to continue to rise as testing continues.

Table 5. Comprehensive Quality Data*

Project	SLOC	Defects**
Flight Systems		
Voyager Period 1	13,950	142
Voyager Period 2	-	-***
Voyager Period 3	-	-
Galileo	27,790	218
Total	41,740	360
Ground Systems		
Voyager Period 1	742,195	200
Voyager Period 2	178,300	1,147
Voyager Period 3	45,300	145
Galileo	1,278,911	2,341
SFOC	342,224	1,038
DSN Mark IV	1,292,715	3,412
Total	3,879,645	8,283

*Adjusted totals. See tables in Appendices B through F for details.

**Defects are defined as problem and operational failure reports
- post development defects
Data is incomplete

Note: At the time of the study, SFOC and the non-NASA project had not completed development

*** " - " indicates data was unavailable

3.2.2 Quality Ratios

Table 6 presents the quality ratio (defects per thousand lines of source code [KSLOC]) for flight and ground systems. For the two flight projects with defect data, there was an average of 8.6 defects per KSLOC left in the code post-development.

The six ground systems had 2.0 defects per KSLOC. As can be seen from Table 6, the number of defects per KSLOC for the six ground systems remains relatively consistent, considering the difficulties encountered in collecting defect data.

But as in the case of productivity, there is a difference in quality figures between flight and ground systems. In this case, however, even though flight systems appear to have greater than four times more defects per KSLOC than ground systems, it may be an artifact in the data. With flight systems there is certainly better defect collection and reporting than with ground software--most likely due to the high visibility in the spacecraft test environment. When the information for this report was collected, SPA staff were repeatedly told by projects that for ground systems many defects were found and fixed but never formally reported or recorded. It is recommended that a system be put in place that records ground defects as well. If defects are not tracked, it is difficult to improve either the process or the product.

Table 6. Quality Ratios

Project	Defects/KSLOC*
Flight Systems	
Voyager Period 1	10.20
Voyager Period 2	- **
Voyager Period 3	-
Galileo	7.80
Cumulative Average* **	**8.60**
Ground Systems	
Voyager Period 1	0.27
Voyager Period 2	6.43
Voyager Period 3	3.20
Galileo	1.80
SFOC	3.03
DSN Mark IV	2.60
Cumulative Average* **	**2.10**

* Data is incomplete
** Defect data not available
*** Adjusted total defects for all projects divided by adjusted total KSLOC for all projects.

Note: Defect data is generally unreliable because there is no standard definition of a post-development defect at JPL and because there is no standard procedure for tracking defects.

3.2.3 Comparable Quality Data

Table 7 presents three non-JPL quality reports that shed some light on the JPL data.

At IBM Systems Programming, the number of ground system defects discovered after delivery ranged from 10 per thousand lines of code in the 1970s to 3 per thousand lines of code in 1980 to 1 per thousand lines of code in 1985. At IBM/FSD Houston, the number of flight system defects released to the customer (in this case NASA) ranged from 2.25 per thousand lines of code in 1982 to .08 per thousand lines of code in 1985. Both IBM Systems Programming and IBM/FSD Houston have rigorous quality programs. They use the technique of Fagan inspections, and they find and fix defects early. Between 1982 and 1985 IBM/FSD went from finding 50% to finding 90% of all defects before configuration control. In contrast, a report developed by the Rome Air Development Center (U.S. Air Force) that covered 59 airborne, strategic and tactical systems representing over five million lines of code reflected a fault density of 9.4 defects per thousand lines of code.

Table 7. Comparable Quality Data

o Average JPL Quality		Defects/KSLOC
- Flight Software		8.6
- Ground Software		2.1
o IBM Systems Programming*		Defects/KSLOC
- Ground Software:	1970	10
	1980	3
	1985	1
o IBM FSD Space Shuttle**		Defects/KSLOC
- Flight Software:	1982	2.25
	1983	0.97
	1984	0.08
o RADC Reliability Report***		
- Systems: 59 (Airborne,Strategic, Tactical,etc.)		
- Lines of Code: 5,235,000		
- Quality: 9.4 Defects/KSLOC		

* A. M. Pietrasanta, "Software Engineering Management", IBM Japan Technology Institute, September 1987

** B. G. Kolkhorst, "Space Shuttle Primary Onboard Software Development: Process Control and Defect Cause Analysis", IBM 92-0069

***Software Reliability Resolution and Estimation Guidebook, RADC-TR-87-171 August 1987

SECTION 4

CONCLUSIONS

The JPL SPA Metrics Study data collection effort was difficult and complex, especially because data was often gathered years after the fact. During the data collection process, weaknesses were discovered in various areas. One overall weakness that SPA discovered was in the recording and tracking of ground software quality (i.e., defects). There appears to be no consistent system in place to accurately report all major ground software defects. Nevertheless, the three workyears expended on this effort produced worthwhile results. JPL now has a rough measurement foundation for software productivity and software quality and an order-of-magnitude quantitative baseline for software systems and subsystems. In other words, JPL has the beginning of a handle on estimating costs for future projects.

Undertaking this study has also resulted in some observations that can benefit anyone conducting metrics studies:

- Start simple - Using four parameters and a rough order-of-magnitude refinement is not too small a starting point.
- Start as soon as possible - It is surprising how quickly data can be lost or forgotten.
- Follow every lead - Collecting data is like assembling a jigsaw puzzle. No lead is too trivial.
- Don't take no for an answer - "No" usually means you haven't asked the right question.
- Don't pursue tangents.
- Make sure data sources can always be identified.
- Don't get discouraged.

When this study began, it seemed as if it might be impossible to retrieve quantitative data on JPL's software productivity and quality. This study shows not only that it was possible, but how to do it.

SECTION 5

FUTURE PLANS

The SPA long-term goal is to integrate the collection of historical data and ongoing project data with future project estimations. More projects will soon be included in the historical database. The existing data will be refined to have greater granularity. For example, data will be distinguished by languages; new, modified and old code; and JPL and contractor productivity. A goal will be to differentiate between costs incurred during different phases of the software development lifecycle. Data will be assembled on both pre- and post-development defects and on which defects were major and which minor. And, based on the work of Basili and others, additional metrics parameters are being added to the four used in this study.

For projects just starting up, SPA is developing a simple set of recommendations for tracking the progress, quality, and productivity of software development projects. These recommendations will be outlined in a guidebook and explained in a SPA training course.

Finally, SPA is evaluating simple, practical cost-estimating algorithms, and intends to adapt several to the JPL environment. The algorithms will be calibrated using historical and ongoing project data. After the tailored algorithms are developed, SPA will provide a guidebook and training courses for JPL project personnel.

BIBLIOGRAPHY

INTERNAL REPORTS

Hill, Robert E. "Voyager Considerations Regarding Operation Software Capability with Respect to the Large Computer Replacement.", IOM 366.15-197 (JPL internal report)

Molko, P. "Galileo Report for the Software Intensive Systems study", IOM GLL-PMM-85-055 (JPL internal report)

EXTERNAL REPORTS

Kolkhurst, B.G. "Space Shuttle Primary Onboard Software Development: Process Control and Defect Cause Analysis.", IBM 92-0069

Peitrasanta, A.M. "Software Engineering Management", IBM Japan Technology Institute, September 1987

"Software Management Indicators", Air Force Systems Command, AFSCP 800-43, January 1986

"Software Quality Indicators", Air Force Systems Command, AFSCP 800-14, January 1987

"Software Reliability Resolution and Estimation Guidebook", RADC-TR-87-171, August 1987

RELATED MATERIAL

"An Approach to Software Cost Estimation," NASA GSFC, SEL 83-001, February 1984

Bailey, J. and Basili, V., "A Meta - Model for Software Development Resource Expenditures," Fifth International Conference on Software Engineering 1981

Boehm, B., Software Engineering Economics, Prentice Hall, 1981

Grady, R.B. and Caswell, D.L., Software Metrics: Establishing a Company-wide Program, Prentice Hall, 1987

McGarry, F. and Card, D., "Studies and Experiments in Software Engineering Lab. (SEL)", Proceedings of the Tenth Annual Software Engineering Workshop, Goddard Space Flight Center, December 1985

Schultz, H., "Software Management Metrics," TR 88-001, MITRE Lab. August 1985

Shooman, M.L., Software Engineering, McGraw-Hill, 1984

APPENDIX A

DEFINITIONS

APPENDIX A

DEFINITIONS

Four types of cost and quality data were collected during this study. These were:

a. Source Lines of Code (SLOC)
b. Software Development Cost in Thousands of Dollars ($K)
c. Software Development Effort in Workmonths (WM)
d. Defect Counts (DEF)

Each of these data types is described below.

1. **Source Lines of Code** (SLOC) - is defined to be a non-blank, non-comment physical line of code. One thousand lines of source code are abbreviated as KSLOC. This study counts only the new lines of source code delivered to operations.

2. **Thousands of Dollars** ($K) - is the technical labor cost of the software development effort, from the start of the project to delivery to operations, in thousands of dollars. An attempt has been made to normalize all dollars to 1987 dollars. (In the case of SFOC and the non-NASA project it was not possible to break out the dollars by year and the numbers presented in this report for those projects are real dollars.) Specifically included in the costs are:

a. Software requirements specification, design, implementation, test, and integration at the subsystem level.

b. System engineering costs. These costs cannot be allocated to individual subsystems, but are included in the total system and project costs.

c. System integration and test costs. Like system engineering costs, these costs cannot be allocated to individual systems, but are included in the system and project totals.

d. Contract management costs. Depending on the project and the nature of the contract, these costs may be allocated to individual subsystems, entire systems, or the project. The data for each individual project must be consulted to determine how contract management costs were counted.

Unless noted otherwise, excluded costs were:

a. Project management costs,
b. Division representative costs,
c. Line management costs,
d. Software Product Assurance costs,
e. Operations and Maintenance costs,
f. Spacecraft testing costs.

3. **Workmonth (WM)** - is the cost in man-months of the software system development effort, from project inception to delivery to operations. It is total number of months expended by software personnel. Both JPL personnel and contractors were included in these totals. Included and excluded costs are the same as those for dollars.

4. **Defects (DEF)** - are the number of software problem reports recorded for each subsystem's software, starting at System Integration and Test. The types of problem reports counted are:

a. Failure Reports (FRs) - written against Mission Operation systems.
b. Problem/Failure Reports (P/FRs) - written against spacecraft systems only.
c. Discrepancy Reports (DRs) - written against DSN systems during operations.
d. Anomaly Reports (ANOM) - Both DSN systems and the non-NASA project wrote Anomaly Reports during System Integration and Test.
e. Incident/Surprise/Anomaly Reports (ISAs) - written against Mission Operations systems or spacecraft systems during operations.
f. Discrepancy/Anomaly Reports (DARs) - written against SFOC software during development.

Unless otherwise specified, subsystem-level unit and integration testing problem reports are not included in the totals.

The following derived ratios are indicators characterizing software quality and productivity.

a. Productivity = Dollars per source line of code ($/SLOC)
b. Productivity = Source lines of code per workmonth (SLOC/WM)
c. Quality = Defects per thousand lines of source code (DEF/KSLOC)

APPENDIX B

VOYAGER

B.1 Voyager Description

B.2 Voyager Data Collection and Results

APPENDIX B.1

VOYAGER DESCRIPTION

The Voyager mission is to gather information about the structure, evolution, and processes of the outer solar system. The timing of the mission reflects the fact that once every 175 years, the outer planets align themselves such that one spacecraft, using gravitational assists, could pass each on its way out of the solar system.

This study has divided the Voyager project into three periods:

Period 1: FY 1971 - FY 1977 - Voyager pre-launch
Period 2: FY 1978 - FY 1982 - Launch through Saturn encounter
Period 3: FY 1983 - FY 1986 - Post-Saturn through Uranus encounter

Voyager was decomposed into three periods because there was new development taking place during each one of the periods listed above. The Voyager project office also organized the financial data in this same way, assigning new project and account numbers to each of the three periods listed above.

Figure B.1 shows the configuration of the Voyager Flight Software (FSW) and Ground Data System (GDS). A brief description of their subsystems follows:

FSW

Attitude and Articulation Control Subsystem (AACS) - The AACS keeps the spacecraft on the desired course and attitude. It also controls the pointing of the instrumentation and high-gain antennas.

Command and Control Subsystem (CCS) - The CCS receives and executes commands uplinked from the ground.

Flight Data Subsystem (FDS) - The FDS collects data from the science and engineering subsystem aboard the spacecraft and transmits it to the ground for further analysis.

GDS

Telemetry Subsystem (TLM) - The TLM receives the bit stream from the spacecraft, synchronizes and decrypts it, and passes it along to the Data Records Subsystem.

Data Records Subsystem (DRS) - The DRS decommutates and catalogs the incoming data, allowing the science investigators and mission operations personnel to selectively view the engineering and experimental returns.

Spacecraft Analysis Subsystem (SAS) - The SAS provides the mission operations personnel the capability needed to monitor the health of the spacecraft.

Mission Sequencing Subsystem (MSS) - The MSS allows construction of command sequences that will be uplinked to the spacecraft. It includes simulation capabilities that allow the effects of such sequences to be evaluated prior to their being transmitted.

Navigation Subsystem (NAV) - The NAV is used to compute the maneuvers that the spacecraft will have to make to reach its target. To achieve the required accuracy, the NAV software models in extremely fine detail the motions of the planets and satellites that the spacecraft will encounter.

Mission Planning Subsystem (MA&E) - MA&E was developed expressly for the purpose of determining the mission trajectories, and was used only prior to launch.

In addition to those subsystems listed above are some activities which are relevant only at the system level; they cannot be allocated among individual subsystems. The FSW includes the Flight Software System Engineering effort. The GDS includes System Engineering and Computer Support activities.

Not included in this study are two GDS subsystems: Command Subsystem (CMD) and Radio Science Subsystem (RSS). The information available for RSS was not accurate enough to include, while the information relating to the CMD was not readily obtainable.

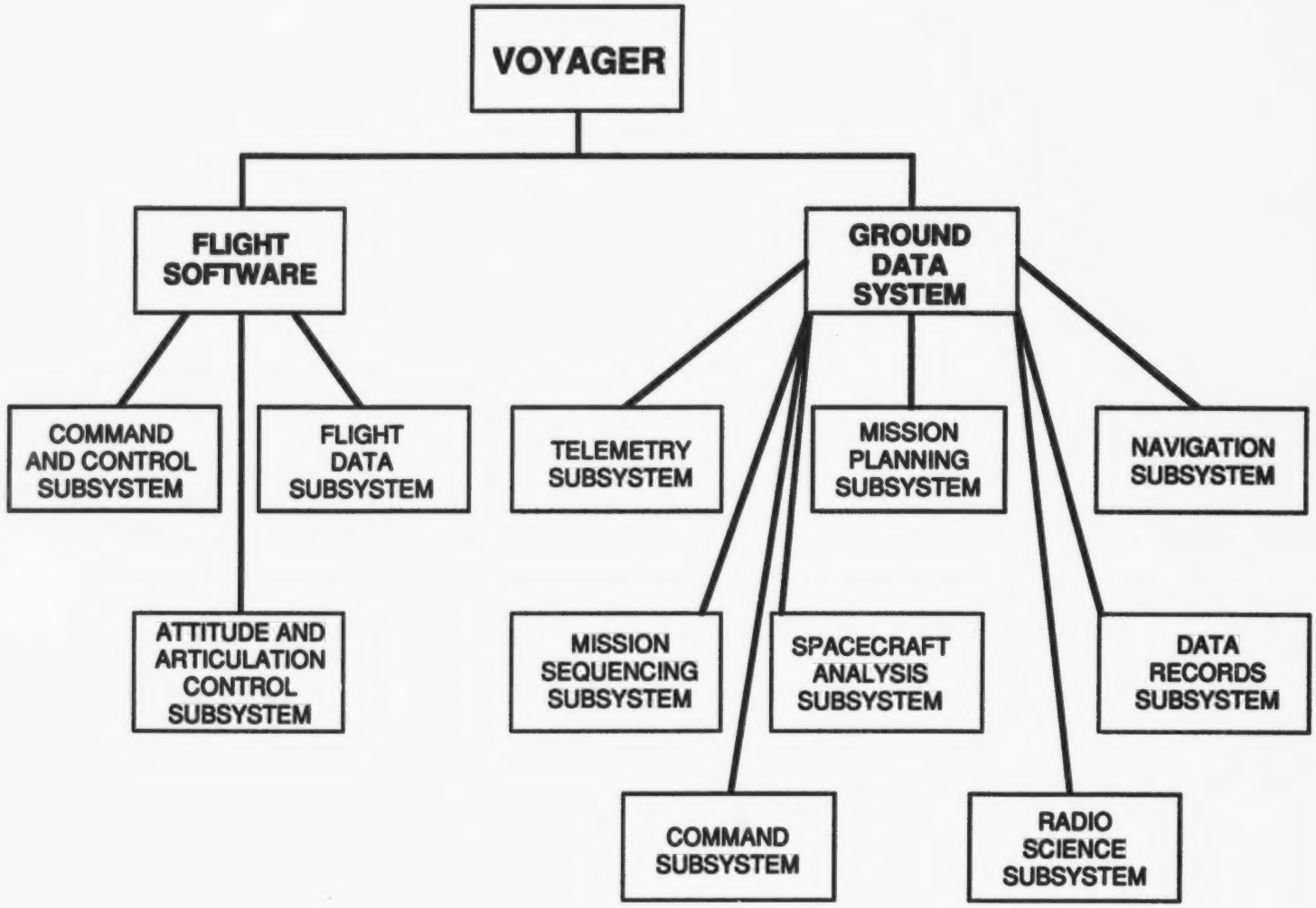


Figure B.1. The Voyager Project System Configuration

APPENDIX B.2

VOYAGER DATA COLLECTION AND RESULTS

The data collected for purposes of this study was obtained from a variety of sources. Following are the primary documents used in the data collection effort and personnel contacted. Unique aspects of the project, data, or collection effort are also discussed.

Dollar data was primarily obtained from the Financial Planning History (D805-13) for Voyager and from talking to project and engineering personnel.

Workmonth data came from Voyager Work Breakdown Structures dating from 1972 to the present and from talking to individuals.

Source lines of code data was found in two documents: D618-327 Voyager Attitude and Articulation Control Subsystem Flight Software Control Document; and IOM 366.15-197 "Voyager Considerations Regarding Operation Software Capability with Respect to the Large Computer Replacement" Robert E. Hill.

Defect data for each subsystem in the GDS has been obtained from the GDS configuration management librarian and the Problem and Failure Accountability (PFA) Center. Defect data for FSW was provided by the Cognizant Development Engineer for Flight Systems.

Personnel Interviewed

$ & WM	Gerry Fleischer
	Sam Deese
	Dick Rice
	Ed Blizzard
	Robert E. Hill
	Joe Stoltzfus
	Paul Penzo
	Joe Beerer
	George Masters
	Pete Breckheimer
	Kerry Erickson
	Doug Griffith

SLOC	(Those listed above and)
	Dick Haga
	Ted Kopf
	Roy Otamura
	Ron Spriestersbach
	Glenn E. Cunningham
	Marilyn Oifer
	Cas Sagoian
	Stuart deJesus
	John Rohr
	Tom Loesch
	Gerry Crichton
	Fred Hammer
	Bob Berwin
	Wyatt Underwood
	Ken Oslund
	Jan Gohlke
	Brent Bennett
DEF	Robert E. Hill
	Pat Laubert
General	Bill Hodgson
	Ed Kelly
	Jim Wilson
	Bob Polansky
	Mike deGyurky
	Ben Toshima
	Dick Rudd
	Bill Gray
	Jack Tupman
	Harry Woo

As has already been noted, there was Voyager data used in this study that had never been permanently recorded. Collection of this data required recall from memory and/or estimation on the part of the cognizant personnel. Some of the data collected covers only a portion of the work effort involved. Metrics involving these numbers represent lower bounds on the quantities being measured and are presented as such. In addition to the above observations, the following limitations should also be considered when viewing the data presented in this report:

1. The cost for the Data Records Subsystem (DRS) is considerably more uncertain than for the other ground software. In particular, during the third period the cost per SLOC of DRS is an order of magnitude less than it was during the first two periods.

2. The costs, staffing, and SLOC data for the MA & E and the TTS are less accurate than those for other subsystems. Development of the MA & E software was less controlled than that for other ground systems, which resulted in fewer records of that development effort

being kept. The TTS development effort was spread over more account numbers than many other subsystems, thereby making it much more difficult to obtain a complete picture of the TTS costs. With regard to the TTS SLOC data, fewer individuals were available who could recall or point to this data.

Table B.1 Voyager Period 1 - Prelaunch

	SLOC	$K	MM	DEF	$/SLOC	SLOC /MM	DEF/ KSLOC
FSW SYSTEM:							
AACS	6000	11535.9	798.0	-+	1922.6	7.5	-
CCS	1950	1683.2	203.6	-+	863.2	9.6	-
FDS	6000	5463.2	200.0	-+	910.5	30.0	-
FSW MISC*				142			
TOTALS & AVG	13950	18682.3	1201.6	142.0	1339.2	11.6	10.2**
GDS SYSTEM:							
DRS	117000	2831.5	243.6	74	24.2	480.3	0.63
MSS	225000	9006.4	602.4	-	40.0	373.5	-
NAV	95000	12423.7	1054.8	30	130.8	90.1	0.32
SAS	52275	1081.1	1168.8	25	20.7	44.7	0.48
MPS	2920	7884.6	75	-	2700.2	38.9	-
TTS	250000	13710.5	400.8	-	54.8	623.8	-
GDS MISC*				71			
TOTALS & AVG	742195.0	46937.8	3545.4	200.00	63.2	209.3	0.27***

- \- Data unavailable
- \+ Data unavailable at the subsystem level but total figure for system is known.
- * Data applying to the overall system that cannot be identified to the subsystem level.
- ** Defect data is used in calculation because it applies to all subsystems (even though DEF/KSLOC figure cannot be calculated for each individual subsystem).
- *** For purposes of this calculation, total defects were assumed to apply to all subsystems.

TABLE B.2 VOYAGER PERIOD 2 - LAUNCH TO SATURN

	SLOC	$K	WM	DEF	$/SLOC	SLOC /WM	DEF/ KSLOC
FSW SYSTEM:							
AACS	-	94.4	-	-	-	-	-
CCS	600	994.3	52.8	-	1657.0	11.4	-
FDS	600	880.1	105.6	-	1466.8	5.7	-
TOTALS & AVG	1200	1968.8	158.4	-	1562*	7.6	-
GDS SYSTEM:							
DRS	80000	1983.6	530.4	116	24.8	150.8	1.45
MSS	90300	3645.4	86.4	5	40.4	1045.1	0.06
NAV	8000	340.5	234	217	42.6	34.2	27.13
SAS	-	3585.6	-	-	-	-	-
MPS	-	2.2	30	51	-	-	-
TTS	-	3125.9	124.8	-	-	-	-
GDS MISC.+				809			
TOTALS & AVG	178300.0	12683.2	1005.6	1198	33**	209.6***	6.43++

- \- DATA UNAVAILABLE
- * LEAVING AACS DOLLARS OUT OF CALCULATION BECAUSE THERE IS NO CORRESPONDING CODE DATA.
- ** LEAVING SAS, MPS, TTS DOLLARS OUT OF CALCULATION BECAUSE THERE IS NO CORRESPONDING CODE DATA.
- *** LEAVING MPS & TTS WORKMONTHS OUT OF CALCULATION BECAUSE THERE IS NO CORRESPONDING CODE DATA.
- \+ DATA APPLYING TO THE OVERALL SYSTEM THAT CANNOT BE IDENTIFIED TO THE SUBSYSTEM LEVEL.
- ++ LEAVING MPS DEFECT DATA OUT OF CALCULATION BECAUSE THERE IS NO CORRESPONDING CODE DATA. FOR PURPOSES OF THIS CALCULATION, MISCELLANEOUS DEFECTS WERE ASSUMED TO APPLY TO TOTAL CODE REPORTED.

Table B.3 Voyager Period 3 - Saturn to Uranus

	SLOC	$K	WM	DEF	$/SLOC	SLOC /WM	DEF/ KSLOC
FSW SYSTEM:							
AACS	-	-	-	-	-	-	-
CCS	600	62.5	7.2	-	104.2	83.3	-
FDS	600	1239.7	190.8	-	2066.2	3.1	-
FSW TOTALS & AVG	1200	1302.2	198.0	-	1085.2	6.1	-
GDS SYSTEM:							
DRS	21000	44.9	4.8	15	2.1	4375.0	0.71
MSS	16300	3057.2	206.4	83	187.6	79.0	5.09
NAV	8000	144.6	159.6	27	18.1	50.1	3.38
SAS	-	3591.9	-	-	-	-	-
MPS	-	-	-	15	-	-	-
TTS	-	1588.5	246	-	-	-	-
GDS MISC.				20			
TOTALS & AVG	45300.0	8427.1	616.8	160.0	71.67*	122.2**	3.20***

- \- Data unavailable

* Leaving SAS and TTS dollar data out of calculation because there is no corresponding code data.

** Leaving TTS workmonth data out of calculation because there is no corresponding code data.

*** Leaving MPS defect data out of calculation because there is no corresponding code data. Miscellaneous defect data IS included because it applies to the overall system.

APPENDIX C

GALILEO

C.1 GALILEO DESCRIPTION

C.2 GALILEO DATA COLLECTION AND RESULTS

APPENDIX C.1

GALILEO DESCRIPTION

The Galileo primary mission is to gather information about the atmosphere of the planet Jupiter. This will include temperature, pressure, and composition data collected by a probe released into the Jovian atmosphere. The Galileo trajectory will also allow a close pass of two asteroids.

The Galileo project began in 1977 and is expected to arrive at Jupiter in 1995, with scheduled completion of the mission in 1997. The data in this report represents the period from inception through FY'86.

Figure C.1 shows the configuration of the Galileo Flight and Ground Software systems and subsystems. A brief description of these systems follows:

Flight Software:

Attitude and Articulation Control Subsystem (AACS) - The AACS determines and controls the attitude of the spacecraft. It also controls the motion of the scan platform on which many of the instruments are mounted.

Command and Data Subsystem (CDS) - The CDS receives, processes, and distributes to other subsystems commands uplinked from the ground. It also collects engineering and scientific data from the spacecraft subsystems and transmits them to the ground.

System Fault Protection (FP) - The FP software is resident on both the above flight subsystems, and was developed by the Systems Engineering organization to allow the spacecraft to recover from situations such as power bus undervoltages, loss of celestial reference, and loss of functionality in a redundant component.

Ground Data System:

Telemetry Subsystem (TLM) - The TLM receives the raw data from the spacecraft. It organizes this data into frames that are ready to be processed by the Data Management Subsystem.

Data Management Subsystem (DMS) - The DMS provides facilities for cataloging the incoming data. It also allows the science investigators and the mission operations personnel to view the information they need to accomplish their tasks.

Mission Sequencing Subsystem (MSS) - The MSS provides the capability of constructing command sequences to be transmitted to the spacecraft. It includes simulation capabilities that predict the result of any command sequences to be transmitted to the spacecraft, thereby minimizing the chances of unexpected changes to the spacecraft's state.

Orbiter Engineering Subsystem (OES) - The OES provides the capabilities used by the mission operations staff in analyzing data received from the spacecraft to monitor its health.

Mission Design Subsystem (MDS) - The MDS software was developed to identify possible spacecraft trajectories that would meet the scientific data collection requirements of the mission.

Navigation Subsystem (NAV) - The NAV is responsible for determining to great accuracy the position and trajectory of the spacecradft during the course of the mission. All maneuver planning is dependent upon the outputs from this subsystem.

In addition to those subsystems listed above are some activities which are relevant only at the system level; they cannot be allocated among individual subsystems. These activities include the Flight Software System Engineering effort and the Ground Software System Engineering effort.

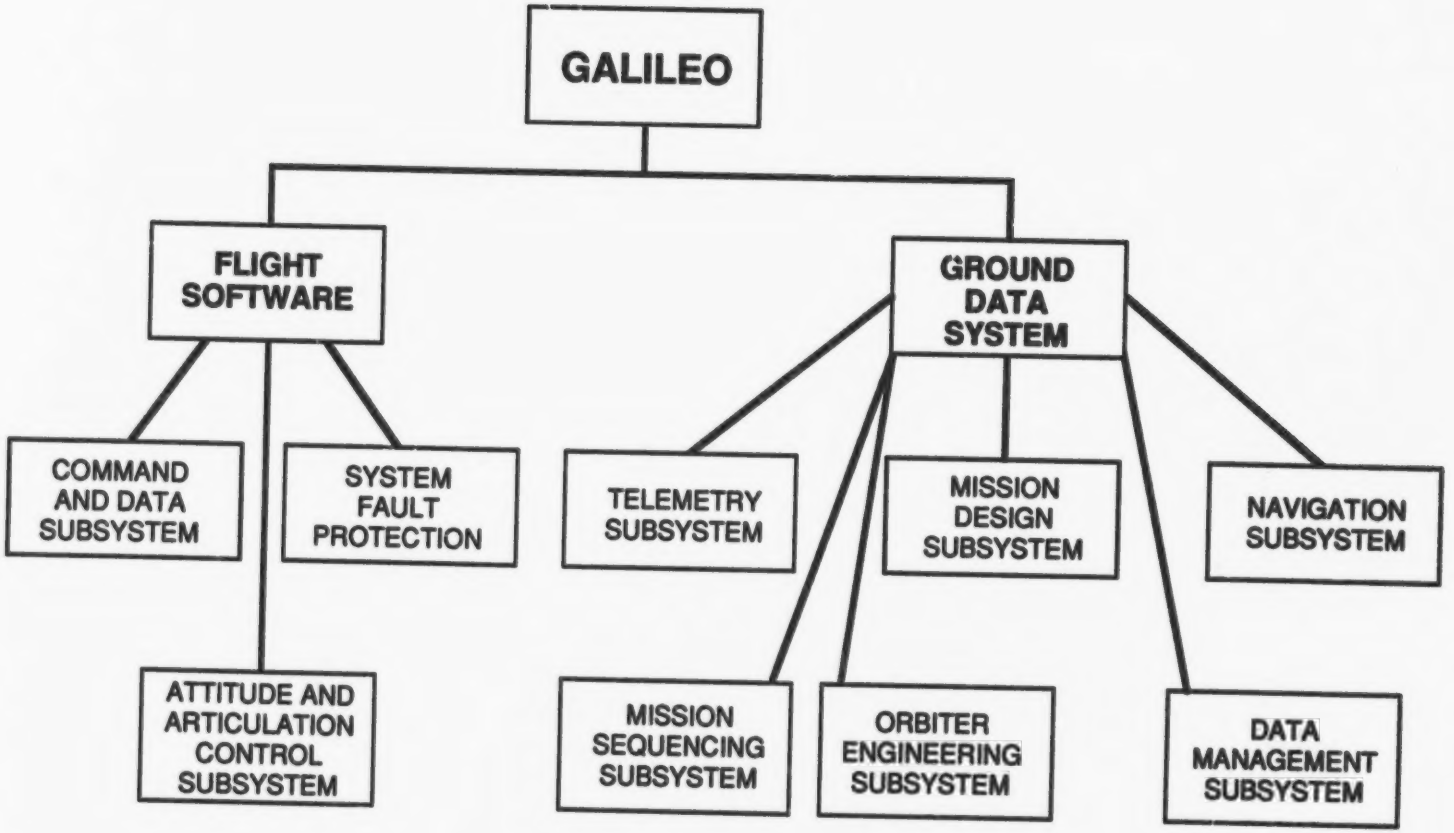


Figure C.1. The Galileo Project System Configuration

APPENDIX C.2

GALILEO DATA COLLECTION AND RESULTS

The data collected for purposes of this study was obtained from a variety of sources. Following are the primary documents used in the data collection effort and personnel contacted. Unique aspects of the project, data, or collection effort are also discussed.

$,WM Backup material for IOM GLL-PMM-85-055 "Galileo Report for the Software Intensive Systems Study" by Pat Molko
- B805, Workforce Planning and Actuals
- C805, Cost Plan History
- D805, Cost and Staffing Actuals
- Financial Planning History Reports

SLOC "Galileo Project Software Margin Management Report" Feb. 19, 1986

Personnel Interviewed

$, WM, SLOC	Chris Hartsough	Wayne Sible
	Bob Mitchell	Jan Chodas
	Pete Breckheimer	John Lai
	Carole Hamilton	Tina Walker
	Pat Molko	R. Haga
	Allen Nikora	L. D'Amario
	Steve Zawacki	Frank Singleton
	Al Schoepke	
DEF	Tonja Harris	

Galileo has suffered many schedule changes throughout the history of the project. One of the most significant delays occurred as a result of the Space Shuttle Challenger disaster in January 1986. Galileo was to have launched soon after but due to the indefinite postponement of the shuttle program as well as concerns about Galileo's propellant system, there were changes made to the overall mission. Many work years and dollars have gone into the project since that time but there have been no major new software deliveries. The existing code has been modified, extensively in some areas, and this kind of information will be examined in later phases of this data collection effort. The data in this report represents the period from inception through FY86.

Table C.1 Galileo Prelaunch Phase

	SLOC	$K	WM	DEF	$/SLOC	SLOC /WM	DEF/ KSLOC
Flight Software System:							
AACS	8290	17780	1510.8	90	2145	5.5	10.9
CDS	19500	10970	1248	128	562	15.6	6.6
FSW MISC.+		111*	27.1*				
TOTALS & AVG	27790	28861	2785.9	218	1039**	10***	7.8
Ground Software System:							
DMS	162945	2872	235.2	244	18	692.8	1.50
MDS	13000	1586	109.2	1	122	119.0	0.08
MSS	399635	15583	1467.6	1033	39	272.3	2.58
NAV	66843	6902	652.3	36	103	102.5	0.54
OES	358000	9303	765.2	209	26	467.9	0.58
TLM	278488	8006	786.6	818	29	354.0	2.94
GDS MISC.+		5843*	617.1*				
TOTALS & AVG	1278911	50095	4633.2	2341	39**	276.0***	1.83

- \- Data unavailable
- \+ Data applying to the overall system that cannot be identified to the subsystem level.
- * System engineering costs and workmonths which cover overall system.
- ** Misc. (System engineering) data is included in dollars total for purposes of calculation.
- *** Misc. (System engineering) workmonth data is included for purposes of calculation.

APPENDIX D

SFOC

D.1 SFOC Description

D.2 SFOC Data Collection and Results

APPENDIX D-1

SFOC DESCRIPTION

The Space Flight Operations Center (SFOC) Project was established in February 1984 and was designed to develop systems that meet Flight Projects' ground data support needs, either with project-specific adaptations or as true multimission capabilities usable by several flight projects. This will help to reduce overall costs of Mission Operations and enable the decommissioning of the aging Mission Control and Computing Center (MCCC) equipment.

The starting point for the extended SFOC Project is the version of the SFOC system that will support Magellan Launch (SFOC Version 7). This report covers all data through Version 7. The software subsystems include core subsystems and applications subsystems.

The SFOC core subsystems provide a "core" capability for a ground data system that allows for basic data transport, data storage and retrieval, and data monitor and display. They are:

Common Data Access Subsystem (CDA), which provides access to data storage and retrieval using the Data Base Management System, file-management, and spooler software; and provides general data management services to other subsystems.

Data Transport Subsystem (DTS), which provides data communications and transport services between and within SFOC subsystems and computers.

Workstation Environment Subsystem (WSE), which provides the data processing environment in which most SFOC applications run, including standard user interface and display.

These three core subsystems provide a standard operating environment in which all SFOC and Flight Project applications will operate.

Several software applications have been developed to support Magellan launch requirements. The design of these application software subsystems furnishes the basis to allow adaptation or conversion for other flight projects included in the scope of the SFOC Project Plan, following the Magellan launch.

Ground Communication Facility Interface (GIF), which provides the interface from the Deep Space Network (DSN) and the Interim Simulation Subsystem to SFOC. GIF will interface with the Ground

Communications Facility Central Communications Terminal (GCFCCT) via a local area network gateway or via router and wideband switch to the SFOC Local Area Networks. GIF includes a data capture and recall funtion.

Telemetry Input Subsystem (TIS), which performs input processing on telemetry frames and DSN monitor blocks, including frame synchronization, error detection and correction (decoding), synchronous and asynchronous extraction, depacketization, and decommutation.

Data Monitor and Display (DMD), which performs standard processing and display of telemetry and other channelized types, including incoming data in real time, near-real time, and recalled from long-term storage.

Digital TV Subsystem (DTV), which generates displays of telemetry and other data for distribution and display via the Closed-Circuit Television Facility (CCTV).

SFOC Monitor and Control Subsystem (SMC), which monitors performance and provides control mechanisms for the SFOC data processing system and specific processes in it.

Central Database (CDB), which retrieves, catalogs, and archives important SFOC data.

Test Workstation (TWS), which provides data stream analysis and troubleshooting capabilities to aid in problem identification and isolation.

Command Subsystem (CMD), which performs real-time command generation and merging with command files prepared by the Sequence Generation Subsystem. It controls transmission of commands to the DSN for radiation to the spacecraft.

Simulation Subsystem (SIM) generates simulated data for test and training purposes and inputs it to SFOC via the GIF Subsystem.

External User Access Subsystem (EUA) provides access with appropriate security to SFOC data.

Magellan High Rate Telemetry Subsystem (MHR) is tasked to process large volumes of telemetry data, which will be shipped to JPL on magnetic tapes from the ground stations that are in radio contact with the spacecraft.

NOTE: CMD, EUA, MHR, and SIM were not delivered in Version 7 (September 16, 1988) and therefore are not included in this report.

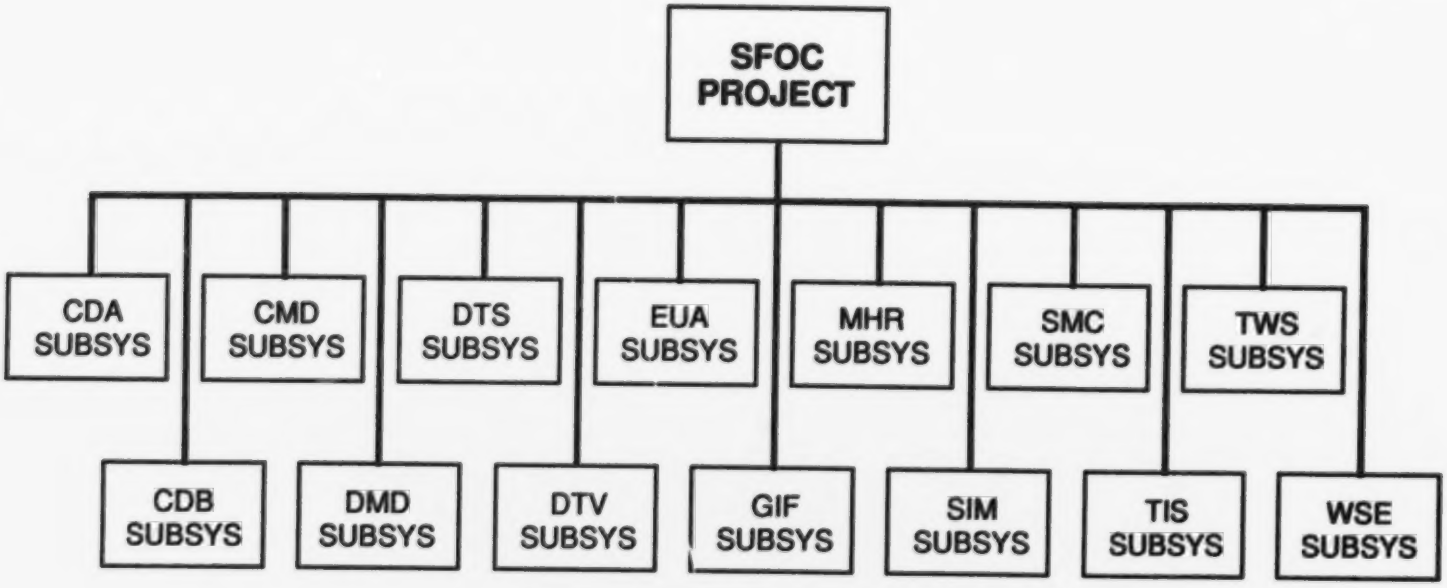


Figure D.1. The SFOC Project System Configuration

APPENDIX D.2

SFOC DATA COLLECTION AND RESULTS

The data collected for purposes of this study was obtained from a variety of sources. Following are the primary documents used in the data collection effort and personnel contacted. Unique aspects of the project, data, or collection effort are also discussed.

Cost and staffing information was obtained from the Resource Status Record (RSR), the D805-13 financial planning history reports kept by Section 367, and from miscellaneous internal reports used in project tracking. These reports contain actual cost and staffing information for each account number to which SFOC charges are made and provide a breakdown by subsystem and version. Section 367 supplied the Work Breakdown Structures (WBS) from which preliminary maps relating accounts to individual systems and subsystems are made. The remainder of the information collected was obtained by meeting with the individuals responsible for the software development.

Source lines of code data was obtained from the system engineers, the subsystem engineers, cognizant software engineers, and cognizant engineers. The data in this section covers the period between the inception of this task in 1984 to September 1988. Cost and staffing information only include the software development resources expended by the development organization (Control Center Data Systems Development Section - 367).

Because SFOC is an on-going task, post-development **defects** are still being discovered. The more complete count of defects reported are development defects, tracked by Discrepancy/Anomaly Reports (DARs). These reports were issued when the subsystem was placed under configuration management. Post-development defects are reported as Failure Reports (FRs) and are under Configuration Management control. Both types of defect data are reported in this report.

Personnel Interviewed

$, WM	Dick Moulder
	Frank Singleton
	Alma Cadwaller
	Hal Norman

SLOC, DEF	Dorothy Huffman Armik Ebrahimian Dave Klemp Marge Craig
General	Mike Ebersole Pam Ray Bob Hall Steve Huffman Chris Hartsough

The data collected for the SFOC project is presented in Table D.1. One ratio that stands out in the table of SFOC data is the productivity figure (SLOC/WM) for WSE. It appears to be very high, both absolutely and relative to other SFOC subystems. According to the Cognizant Engineer, this appears to be due to a combination of factors. WSE consists of libraries and applications and has a presentation layer on top of x-windows, isolating the applications programs from the x-windows; WSE has many small routines; the level of complication was not high; and there was an extremely productive staff that worked well together. In contrast, DTS has a very low productivity figure. This subsystem was very complex and suffered from personnel problems which required redoing the code more than once, lowering productivity.

TABLE D.1 SFOC

SFOC SYSTEM	SLOC	$K*	WM	DEV DEF	POST DEV DEF	$/SLOC	SLOC /WM	DEV DEF/ KSLOC	POST DEV DEF /KSLOC
SMC	25842	743	94.8	105	16	28.8	272.6	4.1	0.62
WSE	66548	621	65.8	217	167	9.3	1011.4	3.3	2.51
CDA	31673	408	46.4	128	50	12.9	682.6	4.0	1.58
CDB	47600	1735	176.2	187	183	36.4	270.1	3.9	3.84
DTS	23881	1421	139.4	157	90	59.5	171.3	6.6	3.77
GIF	20042	614	78.2	118	35	30.6	256.3	5.9	1.75
TIS	47965	1534	165	300	203	32.0	290.7	6.3	4.23
DTV	16016	213	12.8	27	4	13.3	1251.3	1.7	0.25
DMD	40141	814	96.9	203	230	20.3	414.3	5.1	5.73
TWS	22516	451	43.6	109	60	20.0	516.4	4.8	2.66
SFOC MISC**		2768	275.7						
TOTALS & AVG	342224	11322	1194.8	1551	1038	33.1+	286.4++	4.53	3.03

* REAL DOLLARS, 1986-1988

** SFOC SUPPORT/MANAGEMENT COSTS AND WORKMONTHS

\+ MISCELLANEOUS DOLLAR DATA IS INCLUDED BECAUSE IT APPLIES TO ALL SUBSYSTEMS.

++ MISCELLANEOUS WORKMONTH DATA IS INCLUDED BECAUSE IT APPLIES TO ALL SUBSYSTEMS.

APPENDIX E

Non-NASA Project

(Data for this project not available for publication)

APPENDIX F

DSN MARK IV

F.1 DSN MARK IV DESCRIPTION

F.2 DSN MARK IV DATA COLLECTION AND RESULTS

APPENDIX F.1

DSN MARK IV DESCRIPTION

The Deep Space Network (DSN) MARK IV project (1981-1985) is a very large ground based data system providing consolidation of previously collocated Ground Spaceflight and Tracking Data Network stations into the DSN. It provided new monitor and control capability and control size processing to support the Voyager II encounter with Uranus. It also provides automation of the Goldstone, Madrid, and Canberra tracking stations within the DSN.

The DSN MARK IV project consists of four large systems. The system/subsystem structures are shown in Figures F.1 through F.4. A brief description of the systems follows:

Deep Space Communications Complex (DSCC)

- Receiver-Exciter Subsystem (RCV)
- Transmitter Subsystem (TXR)
- Tracking Subsystem (TRK)
- Antenna Mechanical Subsystem (ANT)
- Antenna Microwave Subsystem (UWV)
- Frequency and Timing Subsystem (FTS)
- Technical Facility Subsystem (FAC)
- Telemetry Subsystem (DTM)
- Command Subsystem (DCD)
- Test Support Subsystem (TSA)
- Spectrum Processing Subsystem (DSP)
- Monitor and Control Subsystem (DMC)

The ANT Subsystem includes a 64-meter sub-subsystem, a 34-meter HA/DEC sub-subsystem, and a 34-meter AZ/EL sub-subsystem.

Network Operation Control Center (NOCC)

- Tracking Subsystem (NTK)
- Telemetry Subsystem (NTM)
- Command Subsystem (NCD)
- Monitor and Control Subsystem (NMC)
- Support Subsystem (NSS)
- VLBI Support Subsystem (NRV)
- Radio Science Subsystem (NRS)
- Navigation Subsystem (NAV)

Ground Communication Facility (GCF)

- Central Communications Monitor Subsystem (GCM)
- Data Records Subsystem (GDR)
- Digital Communications Subsystem (GDC)

Test Support System (SPT)

Telemetry System Performance (TLMSPT)
Command System Performance (CMDSPT)
System Performance Executive Test (EXECSPT)
Tracking System Performance (TRKSPT)

Data for NOCC-NAV was unavailable so this subsystem is excluded from this study.

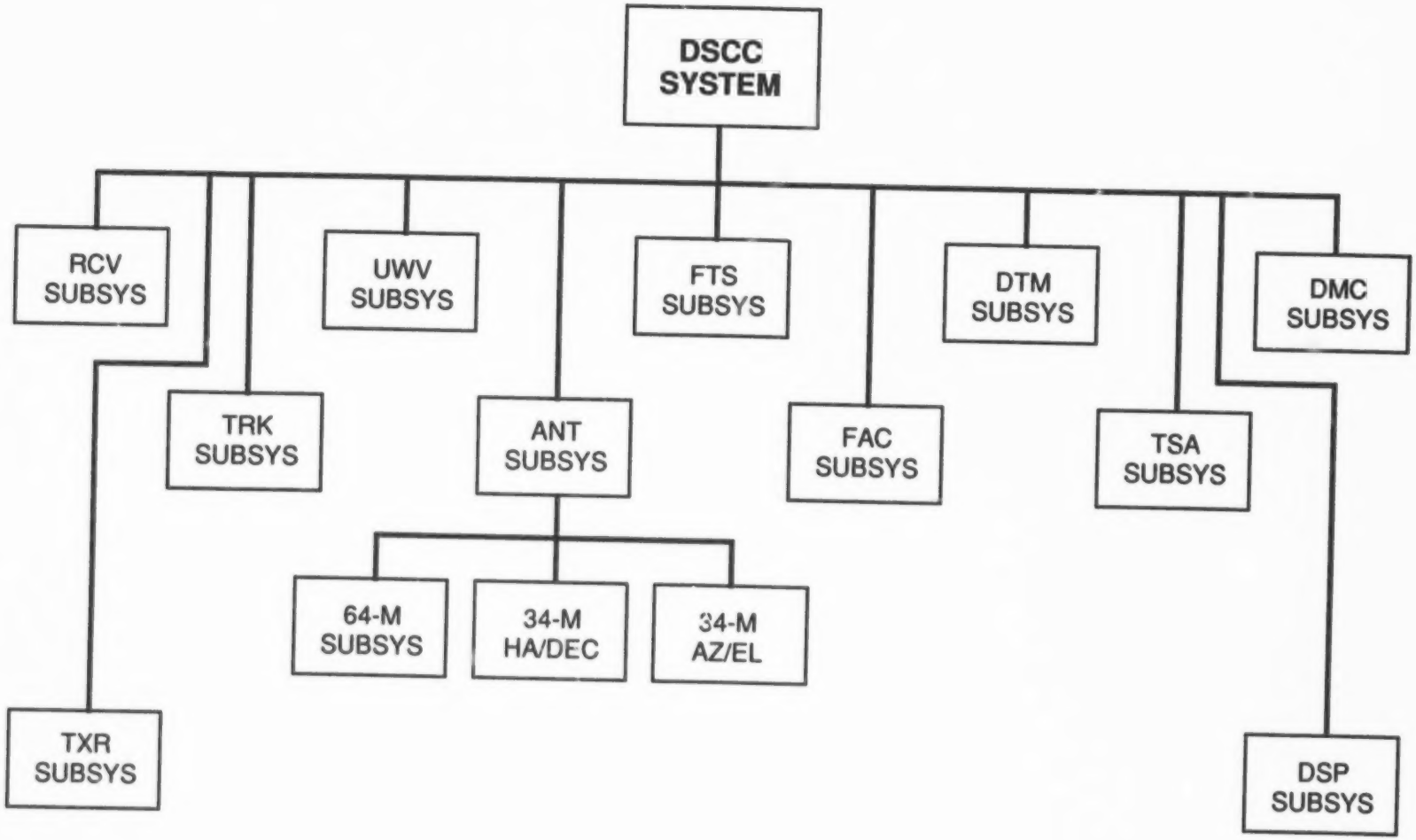


Figure F.1. The DSN MK-IV DSCC System Components

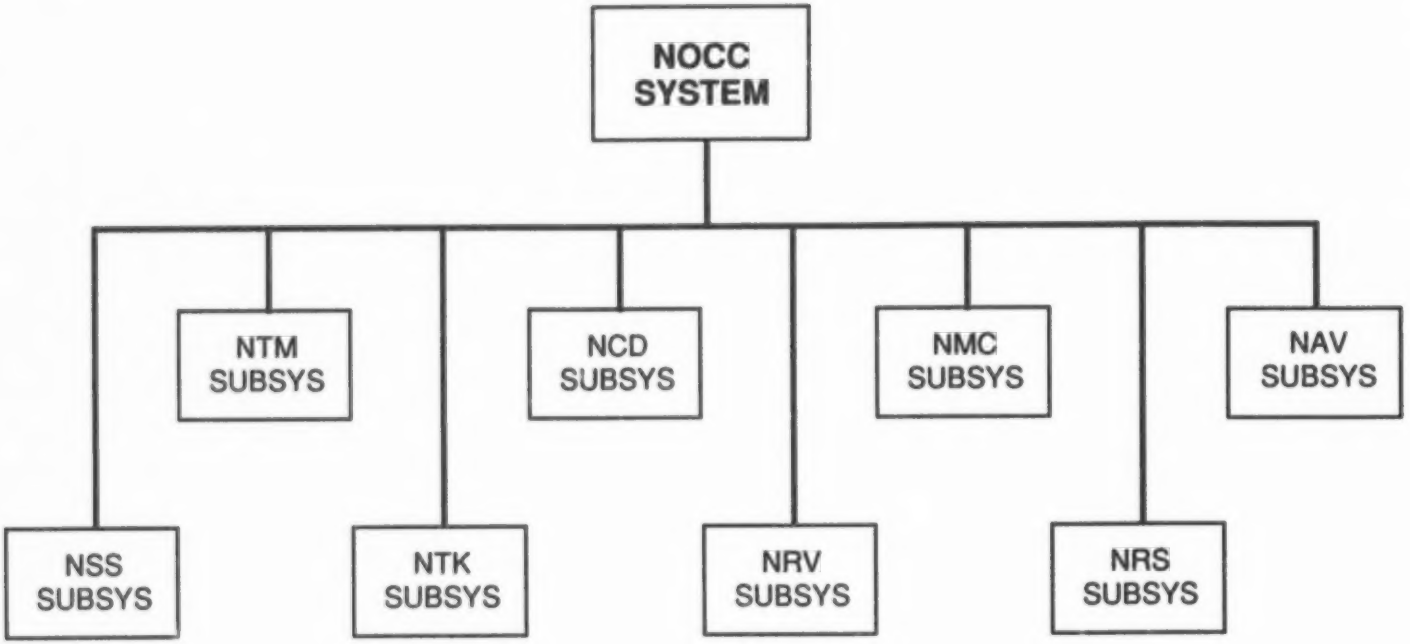


Figure F.2. The DSN MK-IV NOCC System Components

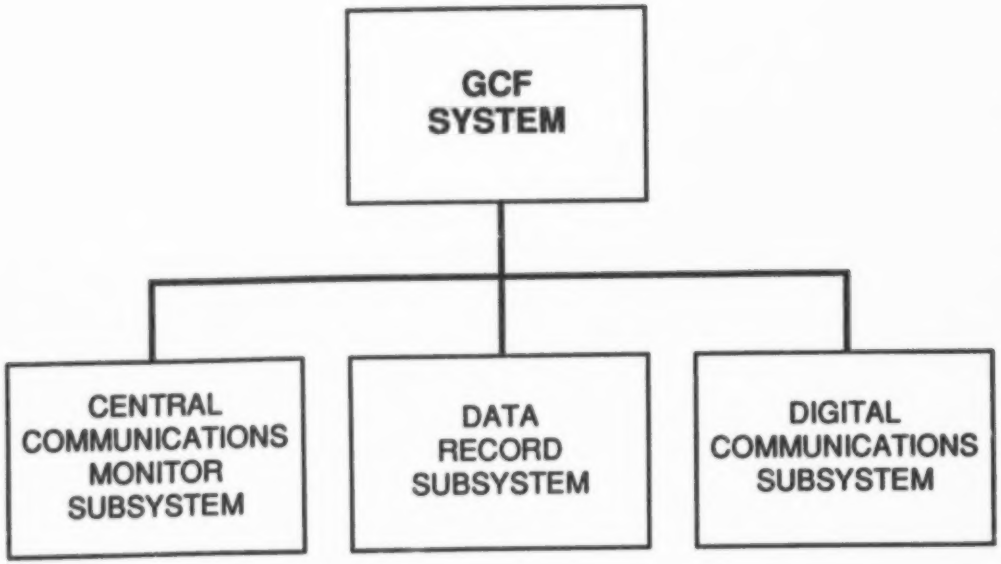


Figure F.3. The DSN MK-IV GCF System Components

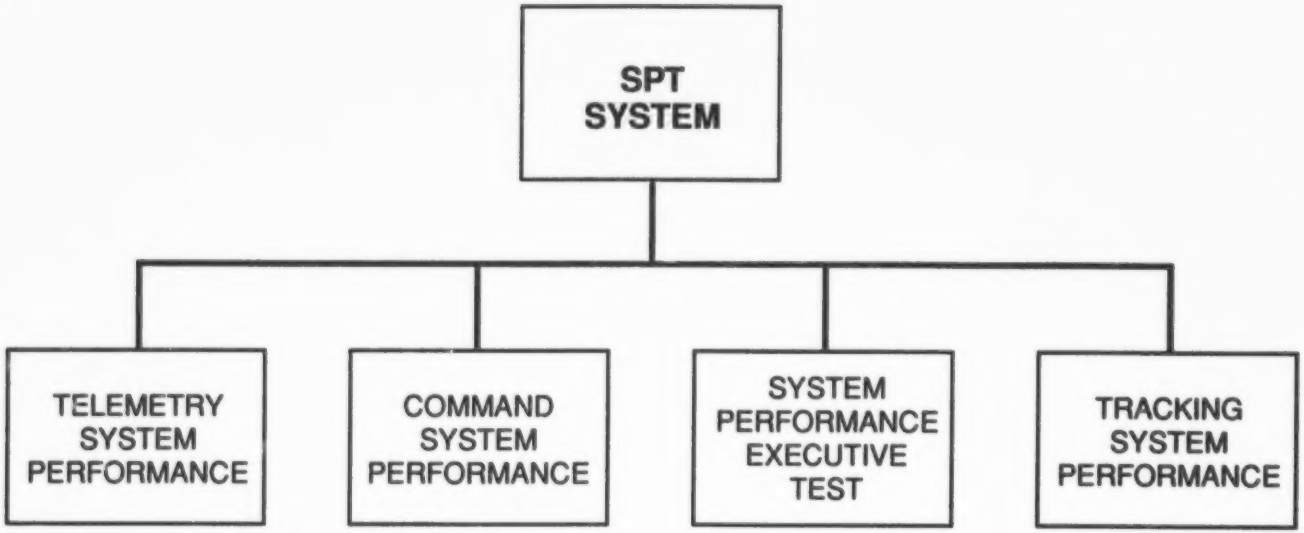


Figure F.4. The DSN MK-IV SPT System Components

APPENDIX F.2

DSN MARK IV DATA COLLECTION AND RESULTS

The data collected for purposes of this study was obtained from a variety of sources. Following are the primary documents used in the data collection effort and the personnel contacted.

Cost and staffing data were obtained from the Tracking and Data Acquisition (TDA) Work Authorization Documents (WADs) and the financial planning history reports for the DSN (B805, Workforce Planning and Actuals; C805, Cost Plan History; and D805, Cost and Staffing Actuals). These documents provided the data for the computation of actual and planned software development costs and staffing for each subsystem. Due to the lack of existence of an accounting map, a great deal of time and effort was expended on extracting this information from these documents which include the account number for all DSN projects for the period 1981-85.

The number of **source lines of code** and **defects** in each subsystem was provided by the Software Planning and Management Center (SPMC), Section 368. The DSN required SPMC to track all software developed on the assembly, subsystem, and system levels during the development, testing, and delivery phases. Although SPMC has provided an accurate assessment of the number of source lines of code and anomalies in the MK-IV software, it does not track Discrepancy Reports (DRs). Some of this information was found in the Release Description Document for each subsystem's software, obtained from the PFA center. However, the remaining DRs were tracked by the contractor (BENDIX) at the stations.

$	Mary Ann Gero
	Joe Dominguez
SLOC/DEF	Pat Shepard
	Mary Wittman
	Ben Parvin
	Rob Warren
	Pete Breckheimer
WM	Chuck Bricker
	Joe Wackley
	Neal Kuo
GENERAL	Paul Westmoreland
	John Leflang

The data collected is presented in Table F.1.

TABLE F.1 DSN MARK-IV

DSN MKIV SYSTEM	SLOC	$K	WM	DEF	$/SLOC	SLOC /WM	DEF/ KSLOC
DSCC SYSTEM							
RCVR	71851	4709.7	738	150	65.5	97.4	2.1
TXR	29147	1543	192	184	52.9	151.8	6.3
TRK	46595	2387.2	278.4	243	51.2	167.4	5.2
ANT-64M	123710	13631.2	645	372	110.2	191.8	3.0
UWV	44527	5822.3	228	-	130.8	195.3	-
FTS	23787	2110	264	4	88.7	90.1	0.2
FAC	12121	1393.7	153.6	2	115.0	78.9	0.2
DTH	60233	1905.5	273.6	540	31.6	220.1	9.0
ANT-34MH/D	14389	1085.2	189.6	54	75.4	75.9	3.8
TSA	38634	1372.2	237.6	129	35.5	162.6	3.3
DSP	70020	1643.1	321.6	159	23.5	217.7	2.3
DMC	162039	7260.9	768	635	44.8	211.0	3.9
ANT-34MA/E	28782	3986	348	-	138.5	82.7	-
DSCC MISC							
TOTALS & AVG	725835	48850	4637.4	2472	67.3	156.5	3.8*
NOCC SYSTEM							
NTK	23703	715.2	86.4	45	30.2	274.3	1.9
NTM	29802	1045.9	145.2	111	35.1	205.2	3.7
NCD	18560	979.7	100.8	54	52.8	184.1	2.9
NMC	50417	-	-	24	-	-	0.5
NRS	31720	1143.7	192	81	36.1	165.2	2.6
NSS	300000	27694.2	2551.2	151	92.3	117.6	0.5
NRV	19855	-	-	24	-	-	1.2
NOCC MISC.				135			
TOTALS & AVG	474057	31578.7	3075.6	625	78.2**	131.3***	0.9+

DSN MARK-IV
(CONTINUED)

	SLOC	$K	WM	DEF	$/SLOC	SLOC /WM	DEF/ KSLOC
GCF SYSTEM							
GCM	21681	1114.6	99.6	14	51.4	217.7	0.6
GDR	38174	1001.5	189.6	188	26.2	201.3	4.9
GDC	69772	1366.7	285.6	90	19.6	244.3	1.3
GCF MISC.							
TOTALS & AVG	129627	3482.8	574.8	292	26.9	225.5	2.3
SPT SYSTEM							
TRKSPT	11534	4349.1	148.8	3	377.1	77.5	0.3
TLMSPT	8965	901.2	109.2	6	100.5	82.1	0.7
CMDSPT	8398	621.2	74.4	2	74.0	112.9	0.2
EXECSPT	7608	-	-	12	-	-	1.6
SPT MISC.							
TOTALS & AVG	36505	5871.5	332.4	23	203.2++	86.9^	0.6
TOTALS & AVG	1366024	89783	8620.2	3412	69.7*	149.4^	2.6*

DSN MARK-IV SUMMARY
FOOTNOTES

* UWV & ANT 34A/E CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING DEFECT DATA.

** NMC & NRV CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING DOLLAR DATA.

*** NMC & NRV CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING WORKMONTH DATA.

\+ MISCELLANEOUS DEFECT DATA IS INCLUDED FOR THIS CALCULATION BECAUSE THE DATA APPLIES TO THE OVERALL SYSTEM.

\++ EXECSPT CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING DOLLAR DATA.

A EXECSPT CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING WORKMONTH DATA.

B NMC, NRV, EXECSPT CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING DOLLAR DATA.

C NMC, NRV, EXECSPT CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING WORKMONTH DATA.

D UWV & ANT 34A/E CODE DATA NOT INCLUDED FOR THIS CALCULATION BECAUSE THERE IS NO CORRESPONDING DEFECT DATA. MISCELLANEOUS DEFECT DATA <u>IS</u> INCLUDED FOR THIS CALCULATION BECAUSE THE DATA APPLIES TO THE OVERALL SYSTEM.